**National Textile University, Faisalabad**

**Department of Computer Science**

|  |  |
| --- | --- |
| **Name:** | Rizwana Bashir |
| **Class:** | BSCS-5th-B |
| **Registration No:** | 23-NTU-CS-FL-1088 |
| **Home Task:** | Week-6 |
| **Submission Date:** | 22-Oct-2025 |
| **Course Name:** | Embedded IoT Systems |
| **Submitted To:** | Sir Nasir Mahmood |

**Week – 6 Home Task**

**Project Title: ESP32 with DHT Sensor, LDR, and OLED Display**

1. **Objective**

The main objective of this project is to design an **IoT-based environmental monitoring system** using an **ESP32 microcontroller**.  
It reads **temperature** and **humidity** from a DHT sensor, measures **ambient light intensity** using an LDR (Light Dependent Resistor), and displays all sensor readings on an **OLED screen** as well as the **Serial Monitor**.

1. **Components Required**

| **Component** | **Description** | **Quantity** |
| --- | --- | --- |
| **ESP32** | Main microcontroller with Wi-Fi and Bluetooth support | 1 |
| **DHT11 or DHT22 Sensor** | Measures temperature and humidity | 1 |
| **LDR (Light Dependent Resistor)** | Measures light intensity | 1 |
| **10kΩ Resistor** | Used as a voltage divider with LDR | 1 |
| **OLED Display (128×64, I2C)** | Displays readings | 1 |
| **Jumper Wires** | For connections | Several |
| **Breadboard** | For assembling circuit | 1 |

1. **Circuit Connections:**

| **Component** | **ESP32 Pin** | **Description** |
| --- | --- | --- |
| **DHT Sensor (Data)** | GPIO 14 | Reads temperature & humidity |
| **LDR** | GPIO 34 (Analog Input) | Reads light intensity |
| **OLED SDA** | GPIO 21 | I2C Data line |
| **OLED SCL** | GPIO 22 | I2C Clock line |
| **VCC (OLED, DHT, LDR)** | 3.3V | Power supply |
| **GND (OLED, DHT, LDR)** | GND | Ground |

1. **Working Principle**
2. The **DHT11/DHT22** sensor measures the **temperature** and **humidity** of the surrounding environment.
3. The **LDR** senses the **light intensity**; as light increases, the resistance of the LDR decreases, producing a higher analog voltage.
4. The **ESP32** reads:
   * Digital data from the **DHT sensor** using the DHT library.
   * Analog voltage from the **LDR** using the analogRead() function.
5. The **OLED display (SSD1306)** shows:
   * Temperature in °C
   * Humidity in %
   * Light intensity (ADC value and voltage)
6. The same readings are also printed on the **Serial Monitor** for debugging and logging.
7. **Complete Source Code:**

#include <Arduino.h>

#include <Wire.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

#include <DHT.h>

// --- Pin Configuration ---

#define DHTPIN 14        // DHT data pin

#define DHTTYPE DHT11    // or DHT22

#define LDR\_PIN 34       // Analog pin for LDR

#define SDA\_PIN 21       // I2C SDA

#define SCL\_PIN 22       // I2C SCL

// --- OLED Setup ---

#define SCREEN\_WIDTH 128

#define SCREEN\_HEIGHT 64

Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, -1);

// --- DHT Sensor Setup ---

DHT dht(DHTPIN, DHTTYPE);

// --- Setup Function ---

void setup() {

  Serial.begin(115200);

  Serial.println("ESP32 DHT + LDR + OLED Project Starting...");

  // Initialize I2C for OLED

  Wire.begin(SDA\_PIN, SCL\_PIN);

  // Initialize OLED Display

  if (!display.begin(SSD1306\_SWITCHCAPVCC, 0x3C)) {

    Serial.println("SSD1306 allocation failed");

    for (;;);

  }

  display.clearDisplay();

  display.setTextColor(SSD1306\_WHITE);

  display.setTextSize(1);

  display.setCursor(0, 0);

  display.println("Initializing...");

  display.display();

  // Initialize DHT Sensor

  dht.begin();

  delay(1500);

}

// --- Main Loop ---

void loop() {

  // --- Read DHT Sensor ---

  float temperature = dht.readTemperature();

  float humidity = dht.readHumidity();

  // --- Read LDR ---

  int adcValue = analogRead(LDR\_PIN);

  float voltage = (adcValue / 4095.0) \* 3.3;

  // --- Check for DHT Errors ---

  if (isnan(temperature) || isnan(humidity)) {

    Serial.println("Error reading DHT sensor!");

    return;

  }

  // --- Print Data on Serial Monitor ---

  Serial.println("===========================");

  Serial.printf("Temperature: %.2f °C\n", temperature);

  Serial.printf("Humidity: %.2f %%\n", humidity);

  Serial.printf("LDR ADC: %d | Voltage: %.2f V\n", adcValue, voltage);

  // --- Display on OLED ---

  display.clearDisplay();

  display.setTextSize(1);

  display.setCursor(0, 0);

  display.println("Hello IoT Project");

  display.setCursor(0, 16);

  display.print("Temp: ");

  display.print(temperature);

  display.println(" C");

  display.setCursor(0, 28);

  display.print("Humidity: ");

  display.print(humidity);

  display.println(" %");

  display.setCursor(0, 40);

  display.print("LDR ADC: ");

  display.println(adcValue);

  display.setCursor(0, 52);

  display.print("Voltage: ");

  display.print(voltage, 2);

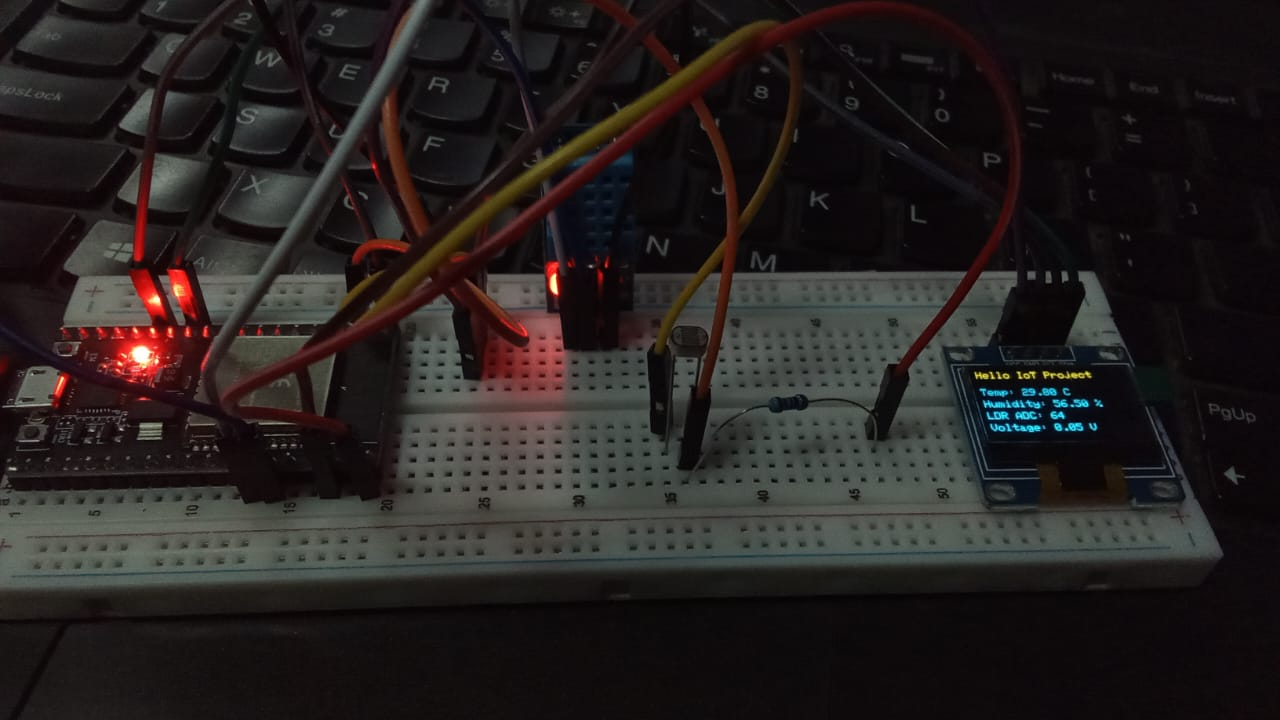
  display.println(" V");

  display.display();

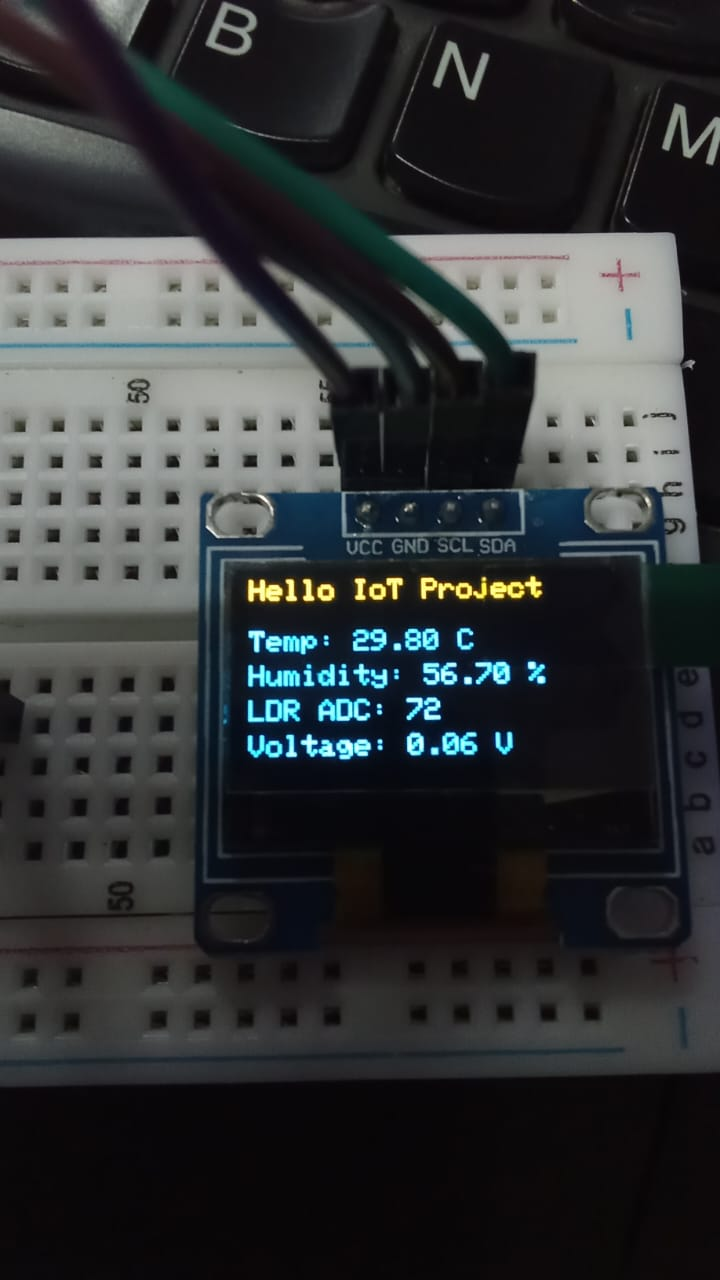
  delay(2000); // Update every 2 seconds

}

1. **Output Description on OLED:**

****

* It shows the temperature , humidity , voltage and LDR ADC digital values on the OLED screen.

****

1. **Explanation of Key Functions**

**1. dht.readTemperature()**

This function reads the temperature value from the DHT sensor (either DHT11 or DHT22).  
It communicates digitally with the sensor and returns the surrounding temperature in degrees Celsius (°C). The value is stored as a floating-point number, allowing decimal precision for accurate measurement.  
If the reading fails or the sensor is not connected properly, the function returns NaN (Not a Number), which can be checked in the program to handle sensor errors safely.

**2. dht.readHumidity()**

This function is used to read the relative humidity of the environment from the DHT sensor.  
The returned value is in percentage (%), representing the amount of moisture present in the air.  
Like the temperature function, it also returns NaN if there is any communication error with the sensor.  
By combining temperature and humidity readings, the ESP32 can provide a clear picture of environmental conditions.

**3. analogRead(LDR\_PIN):**

The analogRead() function reads the analog voltage from the LDR pin connected to the ESP32’s ADC (Analog-to-Digital Converter).  
Since the ESP32 uses a 12-bit ADC, this function returns values from 0 to 4095, where:

* A lower value indicates higher light intensity (brighter conditions).
* A higher value indicates lower light intensity (darker conditions).  
  This value is later converted to voltage to make it easier to interpret in real-world terms.

**4. display.begin():**

This function initializes the OLED display and sets up communication between the ESP32 and the OLED using the I2C protocol.  
It prepares the display for operation by setting its power mode, screen size, and address (usually 0x3C).  
If the display is not properly connected or fails to initialize, the program can detect this and show an error message on the Serial Monitor.

**5. display.print() and display.println():**

Both of these functions are used to print text or numeric values on the OLED screen.

* display.print() writes text on the same line and continues printing right after the last character.
* display.println() prints the text and then moves the cursor to the next line, just like in the Serial Monitor.  
  These functions are essential for formatting and arranging sensor readings neatly on the OLED display.

**6. display.display():**

After printing text or data to the OLED’s internal memory buffer, the information is not immediately visible.  
The display.display() function refreshes the OLED screen to show the updated content.  
It ensures that all the new readings (temperature, humidity, and light level) appear on the display at once in real time.

**7. Serial.printf():**

This function is used to print formatted output to the Serial Monitor.  
It works like printf() in C programming, allowing a mix of text and variable values within a single